Energy-Efficient MPC for biped robots

C. Neves* and R. Ventura**

Institute for Systems and Robotics
Instituto Superior Técnico, Lisbon, Portugal
**E-mail: cneves@isr.tecnico.ulisboa.pt*
***E-mail: rodrigo.ventura@isr.tecnico.ulisboa.pt*

Traditional methods for robotic biped locomotion employing stiff actuation display low energy efficiency and high sensitivity to disturbances. Legged locomotion can be modelled as an hybrid system, where continuous dynamic flows, such as the single or double support stages, are interrupted by discrete jumps, such as heelstrike or lift-off. Traditional control systems are not suited to deal with hybrid systems or with the compliance added by passive elements. A Model Predictive Control (MPC) approach is proposed to deal with the hybrid system dynamics. The controller generates energy efficient gaits for a simulated Simplest Walker (SW) mechanism, tracks the gait trajectories in the presence of sensor noise and small disturbances and is able to adapt to strong and impulsive pushes.

Keywords: Energy Efficiency; Humanoid Robot; Model Predictive Control.

1. Introduction

Our workplaces and households feature new maintenance and leisure systems everyday. One of the most anticipated are humanoid robots that autonomously navigate and interact in human oriented or disaster environments, such as Boston Dynamic's ATLAS[1] or PAL Robotics TALOS.[2]

However, numerous challenges need to be dealt with before humanoid robots become commonplace. Two of the most pressing issues are energy efficiency and gait stability. Following the work of McGeer[3] an array of platforms emerged in order to study passive locomotion[4,5] and develop hardware and control strategies that allowed for increased energy efficiency and stability.[6,7]

Several semi-passive actuators have been proposed[8–10] but typical control algorithms are not suited to deal with complacency or the hybrid nature of biped locomotion.

A Model Predictive Control (MPC) approach that generates and tracks feasible energy-efficient joint and torque trajectories is proposed, based on the work presented by C. Neves and R. Ventura.[11] MPC methods were seldom used for real time biped control due to the heavy computational requirements of the optimization, but some class of problems can be solved efficiently, not only due to the problem structure but also by using warm starts or sub-optimal, yet feasible, solutions. MPC approaches have been used to plan foot placement[12] or ZMP trajectories,[13] leaving the joint trajectories and torques to be solved by inverse kinematics. The controller and Simplest Walker model are presented in the next section and in Section 3 the results are laid out.

2. Model Predictive Control

Model Predictive Control (MPC) [14,15] is a method where constrained optimization of a cost function is used to generate inputs for a dynamical system. The cost function is evaluated at every time step and over a discrete sliding time horizon of N time slots, considering a model of the system. The implementation of a Gait Generator and a Trajectory Follower for a SW model using an MPC approach is detailed in this Section. For this work, YALMIP[16] was used to model the optimization problems and choose the appropriate solver.

The system to be controlled is a Simplest Walker, shown in Fig. 1, a typical case study of passive locomotion in the literature.[4,17] This model represents a compass like mechanism featuring two massless legs of length l connected at the hip by a frictionless joint. Both feet are points with mass m and the upper body weight is condensed into a mass M at the hip. The state of the system is defined as $Q = \begin{bmatrix} q \; \dot{q} \end{bmatrix}^T$, with $q = \begin{bmatrix} \theta & \phi \end{bmatrix}^T$ and where θ is the joint angle between the perpendicular of the ground and the stance leg and ϕ is the joint angle between the stance leg and the swing leg. The following definition also applies:

Definition 2.1. Step Failure - For the simplest walker, a step is considered to fail if:

- ◦ The body falls forward or backward - $|\theta| \leq \frac{\pi}{3}$
- ◦ The robot walks backward - $\dot{\theta} < 0$
- ◦ Swing foot touches ground behind stance foot - $\phi - 2\theta = 0 \wedge \theta < 0$
- ◦ The robot starts running - $F_v = -\cos(\theta)(\dot{\theta}^2 - \cos(\theta - \gamma)) < 0$

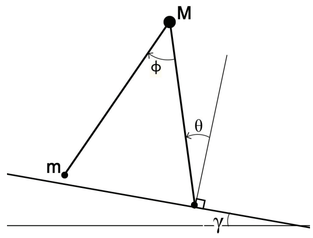

Figure 1. Simplest Walker Model - θ is the angle between the perpendicular to the slope and the stance leg; ϕ is the angle between the stance and swing legs

2.1. *Gait Generator*

The Gait Generator function is to output joint state and torque trajectories $Q = \begin{bmatrix} Q_{(0)} & ... & Q_{(N)} \end{bmatrix}^T$ and $T = \begin{bmatrix} T_{(0)} & ... & T_{(N)} \end{bmatrix}^T$ during one leg swing, considered the flow stage of the hybrid system. While the number of slots is fixed, the corresponding time-step Δt is optimized in order to reach an optimal step period T_s. The slack variable $\Delta s = \begin{bmatrix} s_{(1)} & s_{(2)} \end{bmatrix}^T$ is added to ensure more robust results.

The constraints can be divided into groups: simulation, dynamical and objective constraints. The simulation constraints are

- Assign initial values $Q_{(0)} = Q_0$;
- Assign maximum and minimum values for state $Q_{(k)}$ and torque $T_{(k)}$, where k is a time slot index, and for Δt and Δs.

The dynamical constraints derive from the equations of motion for the passive system, which are well known,[4] and can be adapted for an actuated model to take the form

$$H(q)\ddot{q} + C(\dot{q}, q) + G(q) = T \tag{1}$$

where $H(q)$ is the 2x2 inertia matrix, $C(\dot{q}, q)$ is 2x2 the centripetal and Coriolis forces matrix, $G(q)$ is the 2x1 gravity forces vector and T is a 2x1 vector of the torques applied to the joints. Therefore, the constraints are defined as

$$\ddot{q}_{(k)} = H_{(k)}^{-1}(T_{(k)} - Gm_{(k)} - C_{(k)}) \tag{2}$$

$$Q_{(k+1)} = Q_{(k)} + \begin{bmatrix} \Delta t & \frac{1}{2}\Delta t^2 \\ 0 & \Delta t \end{bmatrix} \dot{Q}_{(k)} \tag{3}$$

$$F_v = -\cos(\theta)(\dot{\theta}^2 - \cos(\theta - \gamma)) \geq 0 \tag{4}$$

where constraints in Eqs. 2 and 3 arise directly from the equations of motion and the constraint in Eq. 4 refrains the system from running - F_v is the vertical force of the stance foot in the ground.

The objective constraints are related to the desired final state. A heel-strike is assured by the constraints in Eq. 5, a desired step length L generates the constraint in Eq. 6 and a desired step period T_s adds the constraint in Eq. 7. Notice that the slack variables will allow for some adjustments to the final objective, but will be penalized in the cost function.

$$\begin{cases} \theta_N > 0 \\ \left\|\phi_{(N)} - 2\theta_{(N)}\right\|^2 = 0 \end{cases} \tag{5}$$

$$\left\|\theta_{(N)} - asin(-L/2)\right\|^2 \leq \Delta s_{(1)} \tag{6}$$

$$N\Delta t - T_s \leq \Delta s_{(2)} \tag{7}$$

The cost function is a quadratic function of the mechanical energy spent in the actuators and the slack variables added in the constraints, in the form

$$J(\dot{q}, t) = G_{torque} \left\|T'.\dot{q}\Delta t\right\|^2 + G_{slack} \left\|\Delta s\right\|^2 \tag{8}$$

where G_{torque} and G_{slack} are gains. The non-linear and non-convex form of the problem requires the use of a non-linear solver, using a Sequential Quadratic Programming (SQP) algorithm.

2.2. *Trajectory Tracker*

The Trajectory Tracker approach consists on solving a Mixed-Integer MPC optimization problem at each time-step over a fixed window of M time-steps. The state and torque trajectories provided by the Gait Generator, Q_P and T_P are used as initial guesses for the Trajectory Tracker, that will optimize state and torque predictions in order to account for external disturbances. As before, the constraints can be divided into simulation, dynamics and objective constraints.

The simulation constraints are:

- Assign initial values $Q_{(0)} = Q_0$;
- Assign maximum and minimum values for state $Q_{(k)}$ and torque $T_{(k)}$, where k is a time slot index, and for Δs.

The dynamic constraints are defined as

$$\begin{cases} \ddot{q}_{(k)} = & H_{(0)}^{-1}(T_{(k)} + T_{ext(k)} - Gm_{(0)} - C_{(0)}) \\ Q_{(k+1)} = & Q_{(k)} + \dot{Q}_{(k)}\Delta t \end{cases} \text{if } hs_{(k)} = 0$$
$$Q^{+}_{(k+1)} = \begin{bmatrix} A_{(q^-)} & 0 \\ 0 & B_{(q^-)} \end{bmatrix} Q^{-}_{(k)} \quad \text{if } hs_{(k)} = 1 \tag{9}$$

where $T_{ext(k)}$ accounts for external disturbances and $hs_{(k)}$ is a binary variable that signals a heelstrike. When a heelstrike is detected, $hs_{(k)} = 1$ and the system performs a state jump, also using known mechanics.[4] Otherwise, the system uses the a linearized model dynamics of the flow stage.

The objective constraints dictate that the predicted trajectory and the optimized trajectory must have a small difference, which is penalized in the cost function:

$$\|q - q_P\|^2 \leq \Delta s_{(1)} \tag{10}$$

$$\|\dot{q} - \dot{q}_P\|^2 \leq \Delta s_{(2)} \tag{11}$$

The cost function of this problem is the same as for the Gait Generator, shown in Eq. 8. The optimization problem is a Mixed Integer Conic problem and the chosen solver was MOSEK.[18]

3. Simulation Results

3.1. *Gait Generator*

Fig. 2 plots the predicted necessary mechanical energy for an actuated simplest walker on a slope $\gamma = 0.004$ to perform a step with length $L = 0.3182$ when initiated on different initial conditions of θ and $\dot{\theta}$, where a darker color represents a higher whole step mechanical energy. The average time of optimization for each set of initial conditions was 5.65 seconds, when using a window of $N = 100$. It should be noted that, similarly to what happens in a passive structure, there is a set of initial conditions that require little actuation to achieve a successful step, which implies that the Gait Generator takes advantage of the natural dynamics of the system.

In order to test how the execution time varies with the window size, several trials were conducted using the same initial conditions and objectives, but for a wide range of window sizes. The evolution of execution time as a function of N is quadratic, as is shown in Fig. 3

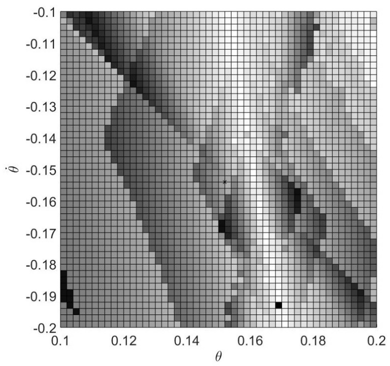

Figure 2. Mechanical Energy needed to perform a step of length $L = 0.3182$ on a $\gamma = 0.004$ slope.

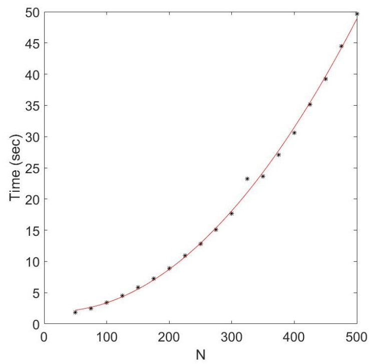

Figure 3. Mean execution time of Gait Generator for different values of N

3.2. *Trajectory Tracker*

The Trajectory Tracker is able to track the trajectories provided by the Gait Generator and adapt to disturbances (modelling sensor noise, complacency and other interferences), performing most iterations in under 0.1s for a $N = 5$ window. However, critical situations such as foot scuffing and heelstrike take up to 3s iterations for the same window and require a reduced $N = 3$ window in order to achieve real time iterations of $\Delta t = 0.15s$.

The sensitivity of the controller to disturbances can be adjusted by tuning the G_{slack} term of the cost function. Simulations show that the system will reduce jittering of the actuators when the cost of the slack is reduced, but it will still react to larger disturbances.

In order to test the reaction to the occurrence of a push forward, a strong impulsive external torque was applied close to the heelstrike, presented in Fig. 5 for $t = 3.88s$, in addition to white noise. The controller responded by advancing the heelstrike and compensating the disturbance over the next step, as is shown in Fig. 4. A second adjustment is required around $t = 5.6s$ in order to avoid a heelstrike during scuffing.

4. Conclusions

This paper describes an MPC approach that is able to generate stable and energy efficient gait and torque trajectories for an actuated simplest walker, with the possibility of tuning the horizon N in order to achieve suitable execution times. While the results focus on walking on a slope for energy comparison, this controller also works for horizontal or upward

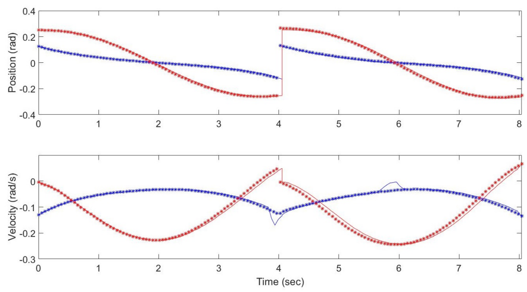

Figure 4. Desired (crosses) and simulated (line) state trajectories

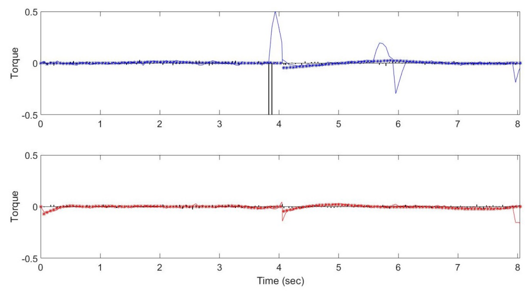

Figure 5. Predicted torque (crosses), external disturbances (bar) and applied torque (line)

slopes. The controller is also able to track the generated trajectories under small constant disturbances, with iterations close to real time, and displays robustness as is able to adapt to a strong push even when applied at a critical time such as before heelstrike.

For future work, the scalability and adaptability of this approach in regards to degrees of freedom and complexity of actuators must be studied.

Acknowledgements

This work was supported by the project FCT [UID/EEA/ 5009/2013] and was partially funded with grant SFRH/BD/80044/2011, also from Fundação para a Ciência e a Tecnologia.

Bibliography

1. S. Kuindersma, R. Deits *et al.*, Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot, *Autonomous Robots* **40**, 429 (2016).

2. O. Stasse, T. Flayols *et al.*, TALOS: A new humanoid research platform targeted for industrial applications (March, 2017).
3. T. McGeer, Passive dynamic walking, *The International Journal of Robotics Research* **9**, 62 (1990).
4. A. Goswami, B. Thuilot and B. Espiau, Compass-like biped robot part i: Stability and bifurcation of passive gaits, *INRIA Research Report N.2996* .
5. M. W. Spong and F. Bullo, Controlled symmetries and passive walking, *IEEE Transactions on Automatic Control* **50**, 1025 (2005).
6. M. Raković, B. Borovac, J. Santos-Victor *et al.*, Biped walking and stairs climbing using reconfigurable adaptive motion primitives, in *IEEE-RAS 16th International Conference on Humanoid Robots*, (2016).
7. M. Brandao, K. Hashimoto, J. Santos-Victor and A. Takanishi, Footstep planning for slippery and slanted terrain using human-inspired models, *IEEE Transactions on Robotics* **32**, 868 (2016).
8. S. Wolf, G. Grioli *et al.*, Variable stiffness actuators: Review on design and components, *IEEE/ASME transactions on mechatronics* **21**, 2418 (2016).
9. C. Neves and R. Ventura, Survey of semi-passive locomotion methodologies for humanoid robots, *15th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Mechanics* , 393 (2012).
10. M. I. Awad, D. Gan, M. Cempini, M. Cortese, N. Vitiello, J. Dias, P. Dario and L. Seneviratne, Modeling, design characterization of a novel passive variable stiffness joint (pvsj), in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (Oct 2016).
11. C. Neves and R. Ventura, Energy efficient MPC for biped semi-passive locomotion, in *Robot 2015 - Second Iberian Robotics Conference*, (2016).
12. M. Morisawa, K. Harada, S. Kajita, S. Nakaoka, K. Fujiwara, F. Kanehiro, K. Kaneko and H. Hirukawa, Experimentation of humanoid walking allowing immediate modification of foot place based on analytical solution, in *Proceedings of IEEE International Conference on Robotics and Automation*, (2007).
13. S. Faraji, S. Pouya, C. G. Atkeson and A. J. Ijspeert, Versatile and robust 3d walking with a simulated humanoid robot (ATLAS): a model predictive control approach, in *Proceedings of IEEE International Conference on Robotics and Automation*, (2014).
14. A. Bemporad and M. Morari, Robust model predictive control: A survey, in *Robustness in identification and control*, (Springer, 1999) pp. 207–226.
15. M. Morari, J. H. Lee, C. Garcia and D. Prett, Model predictive control, *Preprint* (2002).
16. J. Lofberg, YALMIP: A toolbox for modeling and optimization in MATLAB, in *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*,
17. D. G. Hobbelen, *Limit cycle walking* (TU Delft, Delft University of Technology, 2008).
18. MOSEK ApS, *The MOSEK optimization toolbox for MATLAB manual*, (2017).